

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: Bryan G. Yamamoto

Assignee: Mirapoint, Inc.

Title: DUAL-FRAME USER INTERFACE ON GENERIC CLIENT
SOFTWARE

Serial No.: 09/823,425 File Date: March 30, 2001

Examiner: Thong H. Vu Art Unit: 2619

Docket No.: MPT-001

May 20, 2008

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL APPEAL BRIEF

This Appeal Brief responds to the Notification of Non-Compliant Appeal Brief mailed May 12, 2008 and is in support of the Notice of Appeal dated May 5, 2008 and in response to the Fourth Office Action dated January 14, 2008.

/ / / / / / / / /

/

INDEX

I.	REAL PARTY IN INTEREST	3
II.	RELATED APPEALS AND INTERFERENCES	3
III.	STATUS OF CLAIMS	3
IV.	STATUS OF AMENDMENTS	3
V.	SUMMARY OF CLAIMED SUBJECT MATTER	4
VI.	GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL . . .	10
VII.	ARGUMENTS	10
	A. Claims 1-14 and 17-25 are patentable under 35 U.S.C. 102(e) over U.S. Publication 2002/0065848 (Walker) . . .	11
	1. Walker: Overview	11
	2. Claims 1-13 are patentable over Walker . . .	11
	3. Claims 14, 17-25 are patentable over Walker .	18
	B. CONCLUSION	23
VIII.	CLAIMS APPENDIX	24
IX.	EVIDENCE APPENDIX	28
X.	RELATED PROCEEDINGS APPENDIX	28

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Mirapoint, Inc., pursuant to the Assignment recorded in the U.S. Patent and Trademark Office on March 30, 2001 on Reel 011671, Frame 0650.

II. RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no other appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-14 and 17-25 are pending. Claims 15 and 16 are cancelled. Claims 1-14 and 17-25 stand rejected.

In the present paper, rejected Claims 1-14 and 17-25 are appealed.

Pending Claims 1-14 and 17-25 are listed in the Claims Appendix.

IV. STATUS OF AMENDMENTS

All claim amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In accordance with Appellant's invention, generic client software can be configured to resemble a portion of a display window associated with custom client software.

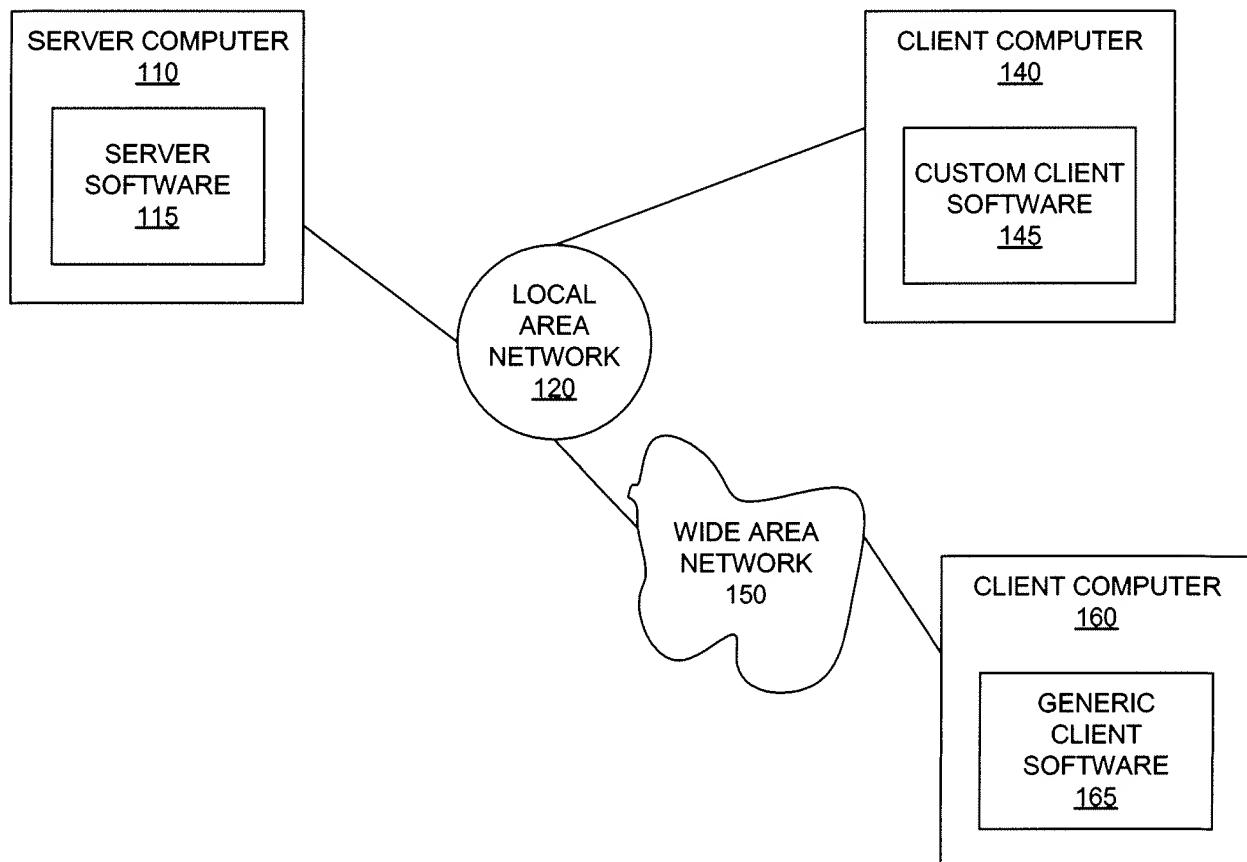


FIGURE 1

As taught by Appellant in the Specification and referring to FIGURE 1 (shown above for convenience),

[0012] Many computer users require access to data records from server software 115 from different computers. For example, an employee may need to access data records from server software 115 while

traveling. For example, in Fig. 1, client computer 160 may be located in a different city than server computer 110 and client computer 140. Access to server computer 110 from client computer 160 is generally limited by the bandwidth of wide area network 150. Furthermore, the specific software such as custom client software 145 may not be available on client computer 160. Therefore, server software 115 is often configured to support use of generic client software 165. In general, generic client software 165 contacts server software 115 and receives computer instructions, which configures generic client software 165 to operate with server software 115 using industry standard protocols such as HTTP and JavaScript.

[0013] However, several issues cause difficulties in replicating the features of custom client software 145 using generic client software 165. One issue is the slow speed of wide area network 150 compared to local area network 120. For example, in most implementations of custom client software 145, all the data identifiers are transmitted to custom client software 145. Thus, custom client software 145 allows a user to easily scan through portions of the data identifiers to locate a desired data record. However, the latency caused by transferring a large list of data identifiers using wide area network 150 may be unacceptable.

[0014] Another issue is due to the static nature of generic client software 165. Specifically, generic client software 165 generally requests specific data pages, such as a web page, from server software 115 using a uniform resource locator (URL). Server software 115 processes the request from generic client software 165 and sends a data page for generic client software to display. The data pages may include links (embedded URLs), which can be selected to request another data page. Thus, for example some web based email systems display a subset of the list of email message headers as links, which can be selected to display a corresponding email message in place of the email message headers. However, conventional configurations of generic client software 165 can not replicate the dual display

areas typical of custom client software 145. Hence, there is a need for a method for configuring generic client software to provide the features of custom client software using industry standard protocols.

To address this issue, Appellant's invention provides for the configuring of generic client software. Specifically, as described in the Specification, paragraph [0015],

generic client software, such as web browsers, is configured to allow different display frames to be synchronized in accordance with one embodiment of the present invention. The synchronization provided by the present invention allows common custom client software features, such as a current data identifier marker and synchronized data list viewing to be implemented. Specifically, in one embodiment of the present invention a data display system is implemented by configuring generic client software. The data display system includes a data display frame and a data list frame. The data display frame is configured to display a current data record. The data list frame is configured to display a set of data identifiers and a current data identifier marker. The current data identifier marker indicates the current data identifier which corresponds to the current data record. The data display system can also include a parent frame that contains both the data display frame and the data list frame, as well as, variables and command scripts for viewing and manipulating the data records.

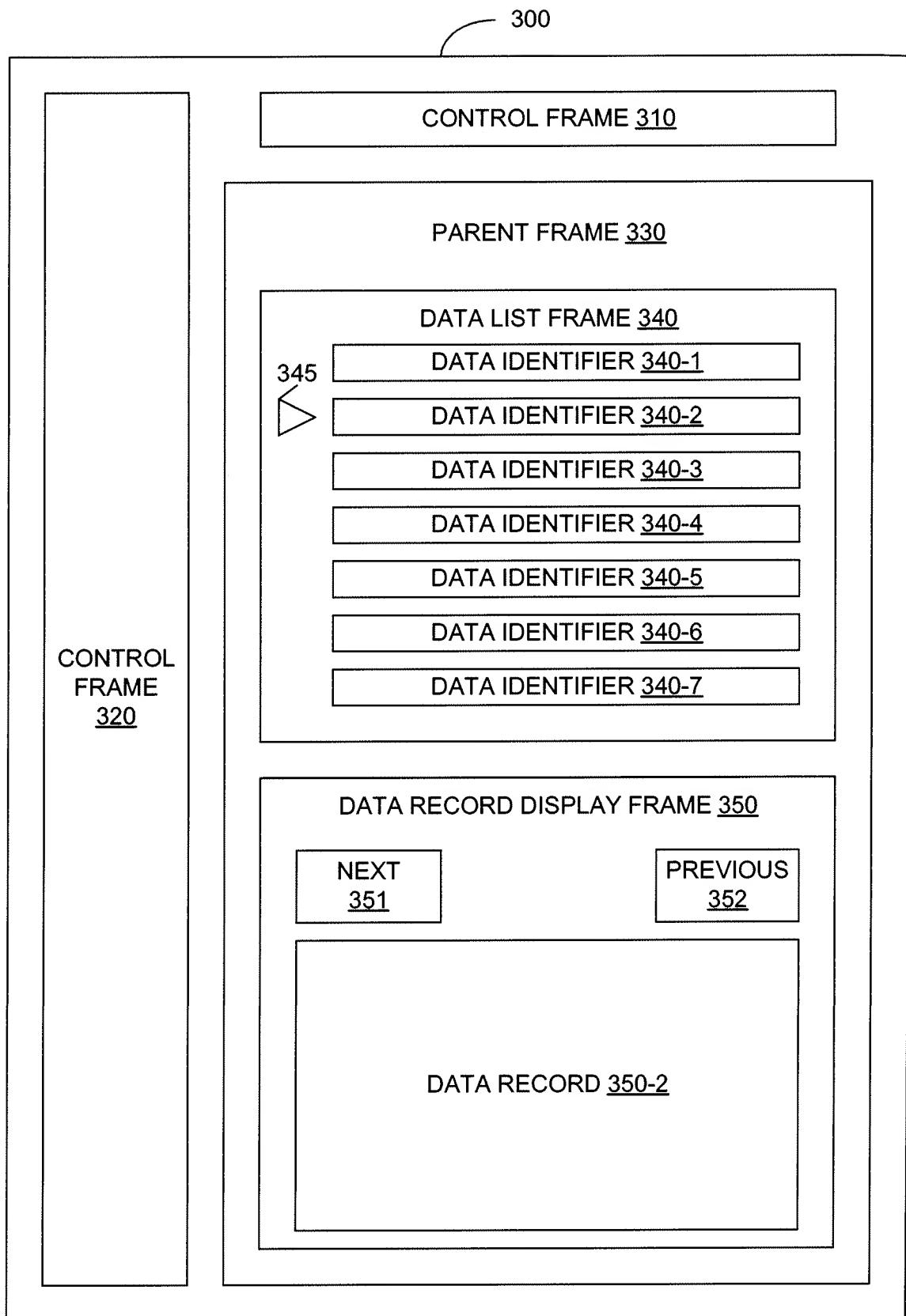


FIGURE 3(a)

As taught by Application in the Specification, paragraph [0027] and referring to Fig. 3(a) (shown above for convenience),

Fig. 3(a) is a display window 300 from generic client software 165 configured in accordance with one embodiment of the present invention. Display window 300 is separated using frames, which segments display window 300 into distinct sections. In general, each frame can be treated as a separate data page. Display window 300 includes a control frame 310, a control frame 320, a parent frame 330, data list frame 340, and data record display frame 350. Parent frame 330... is generally not visible to a user viewing display window 300. ... Display window 300 is made to resemble display window 200 so that a user of custom client software 145 can easily adapt to using generic client software 165 to access server software 115. Thus, control frames 310 and 320 include similar control mechanisms as control areas 210 and 220. Similarly, Data list frame 340 displays data identifiers 340-1 through 340-7 as well as a current data identifier marker 345. The data identifiers in data list frame 340 are a subset of the data identifiers available from server software 115. Different embodiments of the present invention can display greater or fewer numbers of data identifiers in data list frame 340. Because of the bandwidth limitations of wide area network 150, subsets of the data identifiers are sent to generic client software 165 when needed to display.

A concise explanation of the subject matter defined in each of the **independent and dependent claims involved in the appeal** (i.e. **Claims 1, 2, 12, 14, 23, 24**) is provided below. This concise explanation refers to the specification by page and line numbers, and to the drawings by reference numbers.

Claim 1. A data display system implemented by configuring generic client software to resemble a portion of a display window associated with custom client software [**Specification: page 8, lines 17-21; FIGURE 3(a): 300**], the data display system comprising:
a data display frame configured to display a current data record [**Specification: page 9, lines 23-25; FIGURE 3(a): 350**];

a data list frame configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record [**Specification: page 9, lines 1-3; FIGURE 3(a): 340**];

a data display frame lock that indicates whether the data display frame contains a valid data page [**Specification: page 12, lines 29-31; page 13, lines 10-13; FIGURE 5: 550**]; and

a data list frame lock that indicates whether the data list frame contains a valid data page [**Specification: page 12, line 29 to page 13, line 9; FIGURE 5: 540**],

wherein the data display frame and the data list frame facilitate accessing server software over a wide area network, and wherein the data display frame and the data list frame are synchronized over the wide area network using the data display frame lock and the data list frame lock [**Specification: page 14, line 30 to page 15, line 8; page 12, line 29 to page 13, line 13**].

Claim 14. A method of configuring generic client software to synchronize a first frame with a second frame [**Specification: page 10, line 1 to page 13, line 13, FIGURE 5**], the method comprising:

creating a parent frame [**FIGURES 3(a) and 5: 330**] including the first frame [**e.g. FIGURES 3(a) and 5: 340**] and the second frame [**e.g. FIGURES 3(a) and 5: 350**], wherein the first and second frames resemble a portion of a display window created using custom client software [**Specification: page 8, line 27 to page 9, line 22**;

storing a plurality of commands for the first frame and the second frame in the parent frame [**Specification: page 12, lines 21-28, FIGURE 5: 530**];

storing a plurality of variables for the first frame and the second frame in the parent frame [**Specification: page 12, lines 10-20, FIGURE 5: 520**];

displaying a first set of data identifiers in the first frame
[Specification: page 9, lines 1-3; FIGURE 3(a): 340];

displaying a current data record in the second frame
[Specification: page 9, lines 23-25; FIGURE 3(a): 350];

placing a current data record identifier next to a current data identifier corresponding to the current data record
[Specification: page 9, lines 1-3, 23-25; FIGURE 3(a): 340, 350];
and

storing indicators of lock states for the first frame and the second frame in the parent frame **[Specification: page 12, line 29 to page 13, line 9; FIGURE 5: 330, 540, 550],**

wherein storing the indicators of lock states and the plurality of commands and variables allows synchronization of the first and second frames being sent over a wide area network

[Specification: page 12, line 29 to page 13, line 9; FIGURE 5: 540, 550].

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following issues are presented to the Board of Appeals for decision:

(A) Whether Claims 1-14 and 17-25 are patentable under 35 U.S.C. 102(e) over Walker.

VII. ARGUMENTS

A. Claims 1-14 and 17-25 are patentable under 35 U.S.C. 102(e) over Walker.

1. Walker: Overview

As stated in paragraph [0011], Walker teaches a system and method for simultaneously editing of a document by two or more distinct users. The system typically includes one or more workspace servers and a number of client computers. Users access the system by using client software which executes on the client computers.

As stated in paragraph [0013], one or more documents are stored in each workspace. A master copy of each workspace is stored in the file system, which is accessible to the workspace servers. A user can login to the system and connect to a workspace using his client. Each client that is currently connected to a workspace also maintains a local copy of a part or all of the workspace in its client computer. Each workspace contains a container tree data structure and a membership list.

2. Claims 1-13 Are Patentable Over Walker.

Claim 1 recites:

A data display system implemented by configuring generic client software to resemble a portion of a display window associated with custom client software, the data display system comprising:

a data display frame configured to display a current data record;

a data list frame configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record;

a data display frame lock that indicates whether the data display frame contains a valid data page; and

a data list frame lock that indicates whether the data list frame contains a valid data page,

wherein the data display frame and the data list frame facilitate accessing server software over a wide area network, and wherein the data display frame and the data list frame are synchronized over the wide area network using the data display frame lock and the data list frame lock.

a. Discussion of a Wide Area Network

As taught by Appellant, network applications typically refer to computer applications on a first computer that interact with other computer applications running on a second computer. Specification, paragraph [0004]. FIGURE 1 illustrates a server computer 110 with server software 115 that is coupled to a client computer 140 with custom client software 145 via a local area network 120. See, Specification, paragraph [0006]. In turn, local area network 120 is coupled to a wide area network 150. Specification, paragraph [0006]. A client computer 160 is coupled to wide area network 150, typically through a modem or another local area network. Specification, paragraph [0006]. The data bandwidth of local area network 120 is typically between 10 and 100 megabits per second, whereas the data bandwidth between client computer 160 and server computer 110 can be much slower, e.g. 56 kilobits per second. Specification, paragraph [0006].

b. Discussion of Server, Custom Client, & Generic Client Software

Generally, server software 115 runs on server computer 110 to allow custom client software to access the desired data records. Specification, paragraph [0007]. On client computer 140, custom client software 145 is installed to communicate and transfer the desired data records with server software 115. Specification, paragraph [0007]. For example, if the data records are email, server software 115 is a mail server such as Microsoft Exchange and custom client software 145 is a mail client such as Microsoft Outlook. Specification, paragraph [0007]. Generic client software 165 communicates with server software 115 using industry standard protocols, such as HTML and Javascript. Specification, paragraph [0008]. Server software

115 can control generic client software to simulate some of the features of custom client software 145.

c. Discussion of Frames and Synchronization Issues

FIGURE 2 illustrates a simplified display window 200 from custom client software 145. FIGURE 3(a) illustrates a display window 300 from generic client software 165. Notably, display window 300 includes a data list frame 340 and a data record display frame 350. In general, each frame can be treated as a separate data page. Specification, paragraph [0027]. The use of frames 340/350 can be synchronized, thereby allowing generic client software 165 to successfully replicate features of custom client software 145 over wide area network 150.

Synchronization of these frames is difficult because of latency between the network connection between server computer 110 and client computer 160. Specification, paragraph [0030]. That is, server software 115 is generally programmed to assume that all new information from generic client software 165 is sent after receipt of the last data page sent by server software 115. Specification, paragraph [0030]. However, due to network congestion, the last data page may be delayed. Specification, paragraph [0030]. Thus, generic client software 165 (under control of an impatient user) may issue additional requests while using a data page that should have been replaced by the data page sent by server software 115. Specification, paragraph [0030].

Transfer protocol used on the wide area network can also result in loss of synchronization. Specification, paragraph [0031]. Specifically, data transmitted over the wide area network are broken up into small packets and therefore can arrive at the destination (via the wide area network) too slowly

for generic client software 165 to update multiple frames without errors. Specification, paragraph [0031].

The use of parent frame 330 and its constituent components solve the latency and transfer protocol issues associated with synchronization. Specification, paragraph [0032]. For example, referring to FIGURE 5, data list frame lock 540 and data record display frame lock 550 are indicators of whether data list frame 340 and data record display frame 350, respectively, contain valid data pages. Specification, paragraph [0035]. If both data list frame lock 540 and data record display frame lock 550 are in the unlock state, then the appropriate data record from server software 115 is requested. Specification, paragraph [0036]. Otherwise, the request is placed in a command queue 530. Specification, paragraph [0036].

d. Walker Focuses On Updating, Not Synchronization.

Based on the above discussions, Appellant submits that the claim preamble should be construed as if in the balance of the claim. Walker fails to disclose or suggest the recited configuring of the generic client software to resemble a portion of a display window associated with custom client software. Specifically, Appellant submits that Walker focuses on the updating of documents, not the synchronization of the data list frame and the data display frame over the wide area network using the data display frame lock and the data list frame lock.

The Examiner cites paragraphs 0161 and 0382 as teaching the recited data display frame, paragraphs 0059, 0096, 0441, and 0385 as teaching the recited data list frame, paragraphs 0069, 0107, and 0235 as teaching the recited data display frame lock, and paragraphs 0070 and 0114 as teaching the recited data list frame lock. Appellant traverses these characterizations for following reasons.

Paragraph 0161 teaches that there are various default properties for the display of the slides of a document presentation. Paragraph 0161 also teaches that a current_slide field indicates that a live presentation is not currently being conducted using the document presentation. The default properties of a slide and an indicator of a live presentation do not teach the recited data display frame (configured to display a current data record).

Paragraph 0382 teaches that in response to a Refused Lock message, a text editor can display a message that a paragraph is locked by another user and therefore cannot be edited. Alternatively, the user can begin editing the paragraph before a Confirm Lock message is received; however, if a Refused Lock message is subsequently received, then that editing is discarded. The Refused Lock message and the Confirm Lock message do not teach the recited data display frame (configured to display a current data record).

Paragraph 0059 teaches a presentation document that includes three slides: a text slide, an audio/visual clip slide, and a graphic image slide. These slides do not teach the recited data list frame (configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record).

Paragraph 0096 teaches that a *Slide_Article* class is defined to store information relating to slides in a presentation document. This stored information does not teach the recited data list frame (configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record).

Paragraph 0441 teaches that a client marks any local copies of containers in the Skinny Update Message (which is a Post Update Message) as being invalid. This updating does not teach the recited data list frame (configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record).

Paragraph 0385 teaches that a text editor can show unlocked sections, sections locked by a local user, and sections locked by other users in different colors to identify their status. Alternatively, the text editor can display the user's name that locked each section adjacent to or over the section. This color/name identification does not teach the recited data list frame (configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record).

Paragraph 0069 teaches that a container_lock_handle field is used to lock a container and to identify the user that locked that container. This field does not teach the recited data display frame lock (that indicates whether the data display frame contains a valid data page).

Paragraph 0107 teaches a container_lock_handle field has a value of Null, which indicates that the container is not locked by any user. This field (nor any other discussed in 0107) does not teach the recited data display frame lock (that indicates whether the data display frame contains a valid data page).

Paragraph 0235 teaches that a login screen may check a version of a previously-installed client on the client computer. Updates to the client can be performed as necessary. This checking/updating does not teach the recited data display frame lock (that indicates whether the data display frame contains a valid data page).

Paragraph 0070 teaches that a List_Container class maintains a list of child containers and is used to store documents that may be broken down into sections that have a natural "next" and "previous" relationship between them. This class does not teach the recited data list frame lock (that indicates whether the data list frame contains a valid data page).

Paragraph 0114 teaches various fields of a root container. These fields do not teach the recited data list frame lock (that indicates whether the data list frame contains a valid data page).

The Examiner also cites paragraphs 0274 and 0380 as teaching the data display frame and the data list frame being synchronized over the wide area network using the data display frame lock and the data list frame lock. Appellant also traverses these characterizations.

Paragraph 0274 teaches that the terms "freeze" and "frozen" refer to data elements that have been locked by a particular process, entity, or thread for synchronization purposes, whereas the term "lock" refers to containers that have been locked by user for editing. Thus, the locked containers of Walker do not teach the recited limitation in which the data display frame and the data list frame are synchronized over the wide area network using the data display frame lock and the data list frame lock.

Paragraph 0380 teaches that a text editor can be configured to allow a user to lock more than one section of a text document to facilitate editing. This locking does not teach the recited limitation in which the data display frame and the data list frame are synchronized over the wide area network using the data display frame lock and the data list frame lock.

Because Walker fails to disclose or suggest each and every element of Appellant's recited data display system, Appellant

requests reconsideration and withdrawal of the rejection of Claim 1.

Claims 2-13 depend from Claim 1 and therefore are patentable for at least the reasons presented for Claim 1. Based on these reasons, Appellant requests reconsideration and withdrawal of the rejection of Claims 2-13.

3. Claims 14 and 17-25 Are Patentable Over Walker

Claim 14 recites:

A method of configuring generic client software to synchronize a first frame with a second frame, the method comprising:

creating a parent frame including the first frame and the second frame, wherein the first and second frames resemble a portion of a display window created using custom client software;

storing a plurality of commands for the first frame and the second frame in the parent frame;

storing a plurality of variables for the first frame and the second frame in the parent frame;

displaying a first set of data identifiers in the first frame;

displaying a current data record in the second frame;

placing a current data record identifier next to a current data identifier corresponding to the current data record; and

storing indicators of lock states for the first frame and the second frame in the parent frame,

wherein storing the indicators of lock states and the plurality of commands and variables allows synchronization of the first and second frames being sent over a wide area network.

Appellant respectfully submits that the preamble of Claim 14 (i.e. method of configuring generic client software to synchronize a first frame with a second frame) should be construed as if in the balance of the claim for substantially the same reasons presented above for Claim 1. Walker fails to disclose or suggest the recited configuring of the generic

client software to resemble a portion of a display window associated with custom client software. Specifically, Appellant submits that Walker focuses on the updating of documents, not the configuring of generic client software to synchronize first and second frames over a wide area network by storing indicators of lock states and a plurality of commands and variables.

The Examiner cites paragraphs 0156, 0329, 0395, and 0464 as teaching the step of creating, paragraphs 0372-0377 as teaching the step of storing a plurality of commands, paragraph 0103 as teaching the step of storing a plurality of variables, paragraphs 0470 and 0464 as teaching the step of displaying a first set of data identifiers, paragraph 0467 and 0464 as teaching the step of displaying a current data record, paragraphs 0070 and 0369 as teaching the step of placing, and paragraphs 0274, 0380, and 0468 as teaching the step of storing indicators and the wherein limitation. Appellant traverse these characterizations.

Paragraph 0156 teaches that one container can have one child container for each slide. These child containers do not teach the recited first and second frames that resemble a portion of a window created using custom client software.

Paragraph 0329 teaches that a client can create a parent container for a new document. This parent container does not teach the recited first and second frames that resemble a portion of a display window created using custom client software.

Paragraph 0395 teaches how a user can edit a paragraph in a text document or create a new paragraph (by creating a Sibling_Childless.Container). This new paragraph does not teach the recited first and second frames that resemble a portion of a display window created using custom client software.

Paragraph 0464 teaches an editor can open multiple windows for a single document, wherein each window independently displays one or more sections of the document. These multiple windows do not teach the recited parent frame.

Paragraphs 0372-0377 teach configuring a text editor to interpret certain actions by the user as indications that the user wants to edit a document. These actions do not teach the recited step of storing a plurality of commands for the first and second frames in the parent frame.

Paragraph 0103 teaches that independent variables may be defined for each field in a data structure, without formally defining the structural relationship of the fields. These independent variables do not teach the recited step of storing a plurality of variables for the first and second frames in the parent frame.

Paragraph 0470 teaches that a presentation viewer displays the slide corresponding to the current_slide field of the parent article. This current_slide field does not teach the recited step of displaying a first set of data identifiers in the first frame.

Paragraph 0464 teaches that each window can independently display one or more sections of a document. This window does not teach the recited step of displaying a first set of data identifiers in the first frame.

Paragraph 0467 teaches how a presenter can start a presentation using the current_slide field of the parent Presentation_Document. This current_slide field does not teach the recited step of displaying a current data record in the second frame.

Paragraph 0464 teaches how a user can edit a document in a window using an editor incorporated into a client. This editor

does not teach the recited step of displaying a current data record in the second frame.

Paragraph 0070 teaches that a List_Container class can maintain a list of child containers and is used to store documents that may be broken down into sections that have a natural "next" and "previous" relationship between them. This list does not teach the recited step of placing a current data record identifier next to a current data identifier corresponding to the current data record.

Paragraph 0369 teaches that a text editor will display a text document in a window on the display screen of a client computer. When invoked, the text editor begins extracting and displaying the text of the first paragraph, and progressively extracts and displays text (and other elements) from successive sibling containers and their children to fill the display window. This progressive extraction and display do not teach the recited step of placing a current data record identifier next to a current data identifier corresponding to the current data record.

Paragraph 0274 teaches that the terms "freeze" and "frozen" refer to data elements that have been locked by a particular process, entity, or thread for synchronization purposes, whereas the term "lock" refers to containers that have been locked by user for editing. Thus, the locked containers of Walker do not teach the recited step of storing indicators of lock states for the first frame and the second frame in the parent frame, wherein storing the indicators of lock states and the plurality of commands and variables allows synchronization of the first and second frames being sent over a wide area network.

Paragraph 0380 teaches that a text editor can be configured to allow a user to lock more than one section of a text document to facilitate editing. This locking does not teach the recited

step of storing indicators of lock states for the first frame and the second frame in the parent frame, wherein storing the indicators of lock states and the plurality of commands and variables allows synchronization of the first and second frames being sent over a wide area network.

Paragraph 0468 teaches that other than the presenter's client, all other clients that receive a Post_Update message determine that the Post_Update message relates to the start of a presentation. This Post_Update message does not teach the recited step of storing indicators of lock states for the first frame and the second frame in the parent frame, wherein storing the indicators of lock states and the plurality of commands and variables allows synchronization of the first and second frames being sent over a wide area network.

Because Walker fails to disclose or suggest Appellant's recited method of Claim 14, Appellant requests reconsideration and withdrawal of the rejection of Claim 14.

Claims 17-25 depend from Claim 14 and therefore are patentable for at least the reasons presented for Claim 14. Based on these reasons, Appellant requests reconsideration and withdrawal of the rejection of Claims 17-25.

B. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of Claims 1-14 and 17-25 are erroneous, and reversal of these rejections is respectfully requested.

Respectfully submitted,



Jeanette S. Harms
Attorney for Appellant
Reg. No. 35,537

Customer No.: 022888

Telephone: 408-451-5907
Facsimile: 408-451-5908

VIII. CLAIMS APPENDIX

1. (Previously Presented) A data display system implemented by configuring generic client software to resemble a portion of a display window associated with custom client software, the data display system comprising:

a data display frame configured to display a current data record;

a data list frame configured to display a first set of data identifiers and having a current data identifier marker for indicating a current data identifier corresponding to the current data record;

a data display frame lock that indicates whether the data display frame contains a valid data page; and

a data list frame lock that indicates whether the data list frame contains a valid data page,

wherein the data display frame and the data list frame facilitate accessing server software over a wide area network, and wherein the data display frame and the data list frame are synchronized over the wide area network using the data display frame lock and the data list frame lock.

2. (Original) The data display system of Claim 1, further comprising a parent frame containing the data display frame and the data list frame.

3. (Original) The data display system of Claim 2, wherein the parent frame further comprises a plurality of command scripts.

4. (Original) The data display system of Claim 3, wherein the data display frame further comprises a next button associated with a next command script from the plurality of command scripts.

5. (Original) The data display system of Claim 4, wherein the data display frame further comprises a previous button associated with a previous command script from the plurality of command scripts.

6. (Previously Presented) The data display system of Claim 4, wherein the next command script is configured to request a new current data record.

7. (Original) The data display system of Claim 6, wherein the next command script is also configured to update the current data identifier marker.

8. (Original) The data display system of Claim 6, wherein the next command script is also configured to request a second set of data identifiers when the current data record corresponds to a last data identifier in the first set of data identifiers.

9. (Original) The data display system of Claim 1, wherein the current data identifier marker is an arrow.

10. (Original) The data display system of Claim 1, wherein the current data identifier marker is signified by highlighting the current data identifier.

11. (Original) The data display system of Claim 1, wherein the data list frame includes a set of status markers for the set of data identifiers.

12. (Original) The data display system of Claim 1, wherein the data display system is an email client.

13. (Original) The data display system of Claim 1, wherein the generic client software is a web browser.

14. (Previously Presented) A method of configuring generic client software to synchronize a first frame with a second frame, the method comprising:

creating a parent frame including the first frame and the second frame, wherein the first and second frames resemble a portion of a display window created using custom client software;

storing a plurality of commands for the first frame and the second frame in the parent frame;

storing a plurality of variables for the first frame and the second frame in the parent frame;

displaying a first set of data identifiers in the first frame;

displaying a current data record in the second frame;

placing a current data record identifier next to a current data identifier corresponding to the current data record; and

storing indicators of lock states for the first frame and the second frame in the parent frame,

wherein storing the indicators of lock states and the plurality of commands and variables allows synchronization of the first and second frames being sent over a wide area network.

15. (Cancelled)

16. (Cancelled)

17. (Previously Presented) The method of Claim 14, further comprising highlighting the current data identifier corresponding to the current data record.

18. (Previously Presented) The method of Claim 14, further comprising displaying a set of status markers corresponding to the set of data identifiers in the first frame.

19. (Previously Presented) The method of Claim 14, wherein the plurality of commands includes a next command.

20. (Original) The method of Claim 19, further comprising requesting a new current data record when the next command is activated.

21. (Original) The method of Claim 20, further comprising, updating a current data identifier marker when the next command is activated.

22. (Original) The method of Claim 20, further comprising requesting a second set of data identifiers when the next command is activated and the current data record corresponds to a last data identifier in the first set of data identifiers.

23. (Original) The method of Claim 14, wherein the first frame is configured to display a list of email headers.

24. (Original) The method of Claim 23, wherein the second frame is configured to display an email.

25. (Original) The method of Claim 14, wherein the generic client software is a web browser.

IX. EVIDENCE APPENDIX

(NONE)

X. RELATED PROCEEDINGS APPENDIX

(NONE)